

Αναφορά Ανάπτυξης Project: The Best Banking System

Ομάδα Φοιτητών Τμήματος Μηχανικών Πληροφορικής ΔΙΠΑΕ Σερρών

Μάριο Βέλα
Θεοδώσης Στάμογλου
Άγγελος Κουρουγένης

Μάιος 2025

Περιεχόμενα

1	Εισαγωγή και Motivation	2
2	Στόχοι και Εύρος του Project	2
3	Αρχιτεκτονική Συστήματος	2
4	Τεχνολογίες και Βιβλιοθήκες	4
5	Κώδικας και Υλοποίηση	4
6	Αποτελέσματα και Demo	4
7	Σύγκριση με Κώδικα από AI	5
8	Συμπεράσματα και Lessons Learned	5
9	Μελλοντικές Κατευθύνσεις	5

1 Εισαγωγή και Motivation

Το project **The Best Banking System** είναι μια εκπαιδευτική αλλά και πλήρως λειτουργική προσομοίωση ενός τραπεζικού πληροφοριακού συστήματος. Η εφαρμογή καλύπτει βασικές και προηγμένες λειτουργίες μιας σύγχρονης ηλεκτρονικής τράπεζας.

Το πρόβλημα που αντιμετωπίζει είναι η ανάγκη για ασφαλή, επεκτάσιμη και εκπαιδευτική πλατφόρμα διαχείρισης λογαριασμών και συναλλαγών με έμφαση στην προστασία του χρήστη.

Η επιλογή του αντικειμένου έγινε με γνώμονα:

- την ενίσχυση δεξιοτήτων στον αντικειμενοστραφή προγραμματισμό,
- τη χρήση τεχνολογιών ασφάλειας όπως OTP και hash passwords,
- την επαφή με real-world concepts όπως τόκοι, verification και επικοινωνία με email.

2 Στόχοι και Εύρος του Project

Κύριοι στόχοι

- Ανάπτυξη πλήρους C++ backend για λογαριασμούς και συναλλαγές
- Επαλήθευση χρηστών μέσω email OTP
- Αποστολή HTML email με πληροφορίες τόκων
- Αυτόματη εφαρμογή τόκων 2 φορές το χρόνο
- Διαχείριση στοιχείων χρήστη με προστασία δεδομένων

Επέκταση και Εύρος

- Σύνδεση C++ με Python scripts μέσω system calls
- Εφαρμογή πολιτικής επαναπροσπάθειας και κλειδώματος λογαριασμών
- Δυνατότητα πολλαπλών μενού για διαχείριση (User, Account Management, Transactions)
- Υποστήριξη αρχείων '.txt' για αποθήκευση χωρίς βάση δεδομένων
- Modular design με δυνατότητα μελλοντικής σύνδεσης με Web Interface

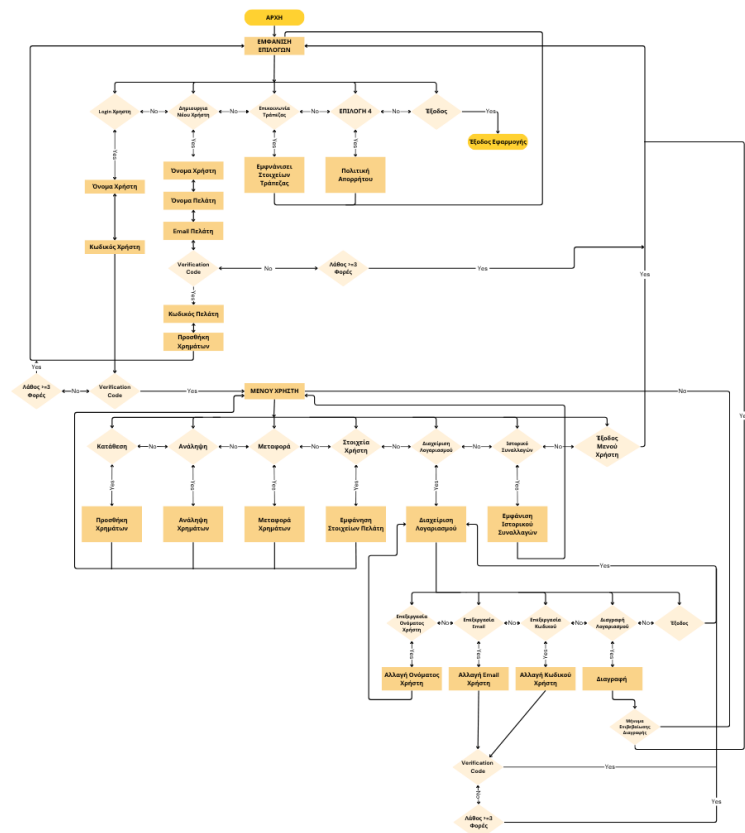
3 Αρχιτεκτονική Συστήματος

Επισκόπηση

Το σύστημα αποτελείται από 3 βασικά στρώματα:

- **Εφαρμογή C++:** Κύριος κορμός λογικής και διαχείρισης δεδομένων
- **Scripts Python:** Αποστολή OTP και ενημερωτικών email (SMTP/SSL)
- **Αρχεία δεδομένων:** Αποθήκευση λογαριασμών και ιστορικού

Workflow Diagram



Σχήμα 1: Workflow Diagram

4 Τεχνολογίες και Βιβλιοθήκες

Γλώσσες και Εργαλεία

- C++ για αντικειμενοστραφή προγραμματισμό
- Python 3 για email handling
- G++ compiler για build
- Git για version control

Βιβλιοθήκες

C++: vector, string, fstream, iomanip, regex, chrono, map, random

Python: smtplib, ssl, email.mime

5 Κώδικας και Υλοποίηση

Η υλοποίηση βασίζεται σε:

1. **banking_system.cpp**: Κύριο αρχείο με κλάσεις BankAccount, Transaction, και BankSystem
2. **send_otp_email.py**: Αποστολή OTP σε πραγματικό χρόνο
3. **send_interest_info_email.py**: HTML ενημέρωση για τους τόκους

Σημαντικά χαρακτηριστικά

- Έλεγχος εγκυρότητας username και password με regex
- Δημιουργία μοναδικού λογαριασμού με prefix-based αριθμούς
- Ενσωμάτωση 2FA με OTP και χρονικό περιορισμό
- Υπολογισμός τόκων (3%) και προσθήκη σε ισολογισμό
- Καταγραφή συναλλαγών με timestamp και file output

6 Αποτελέσματα και Demo

Πραγματοποιημένα Features

- Εγγραφή χρήστη με OTP (έλεγχος σε πραγματικό Gmail)
- Δημιουργία ασφαλών συναλλαγών
- Ιστορικό συναλλαγών ανά χρήστη
- Δομή μενού για ενίσχυση UX
- Μηχανισμός κλειδώματος λογαριασμού σε περίπτωση επαναλαμβανόμενης αποτυχίας

Έλεγχοι & Test Cases

- Εσφαλμένοι κωδικοί: 3 προσπάθειες και κλείδωμα
- OTP λήγει μετά από 60 δευτερόλεπτα
- Το email επιβεβαιώνεται πριν δημιουργηθεί λογαριασμός
- Εφαρμογή τόκων

7 Σύγκριση με Κώδικα από ΑΙ

Πλεονεκτήματα Χειροκίνητης Ανάπτυξης:

- Απόλυτος έλεγχος σε κάθε συνάρτηση και class
- Τεκμηρίωση, επεξήγηση, debugging και καλύτερη κατανόηση
- Δυνατότητα refactor και modular επεκτασιμότητας

Προβλήματα AI-generated code:

- Αγνοούνται corner cases
- Μη αξιόπιστη διαχείριση σφαλμάτων
- Δεν ακολουθεί απαραίτητα design principles

8 Συμπεράσματα και Lessons Learned

- Η συνεργασία γλωσσών (C++ με Python) μπορεί να γίνει ομαλά με καλό σχεδιασμό
- Η ασφαλής αυθεντικοποίηση είναι εφικτή χωρίς βάση δεδομένων
- Τα HTML email βελτιώνουν την εμπειρία χρήστη και αξιοπιστία
- Ο modular σχεδιασμός βοηθά στο testing και future-proofing
- Το σύστημα αποτελεί καλή βάση για πλήρες web banking platform

9 Μελλοντικές Κατευθύνσεις

- Δημιουργία Web Interface με Flask ή React
- Εγκατάσταση SQLite για πιο σταθερή αποθήκευση δεδομένων
- Υποστήριξη IBAN validation
- Role-based access για admins/users
- Αποστολή ειδοποιήσεων για ύποπτες συναλλαγές